

Affect and machine design: Lessons for the development of autonomous machines

by D. A. Norman
A. Ortony
D. M. Russell

Human beings have evolved a rich and sophisticated set of processes for engaging with the world in which cognition and affect play two different but equally crucial roles. Cognition interprets and makes sense of the world. Affect evaluates and judges, modulating the operating parameters of cognition and giving a warning about possible dangers. The study of how these two systems work together provides guidance for the design of complex autonomous systems that must deal with a variety of tasks in a dynamic, often unpredictable, and sometimes hazardous environment.

Animals and humans have two distinct kinds of information processing mechanisms: *affect* and *cognition*. Cognitive mechanisms—mechanisms that interpret, understand, reflect upon, and remember things about the world—are reasonably well understood. But there is a second set of mechanisms, equally important and inseparable—the system of affect and emotion that rapidly evaluates events to provide an initial assessment of their valence or overall value with respect to the person: positive or negative, good or bad, safe or dangerous, hospitable or harmful, desirable or undesirable, and so on.

Although affect and cognition are conceptually and to some degree neuroanatomically distinct systems, from a functional perspective they are normally deeply intertwined. They are parallel processing systems that require one another for optimal functioning of the organism. There is some evidence¹ that people with neurological damage compromising

their emotional (affective) systems become seriously limited in their ability to organize their day-to-day lives, even while appearing to perform normally on a battery of standardized cognitive tasks. They become ineffective actors in a complex world. Furthermore, psychologists and others interested in artificial intelligence have repeatedly urged that affect is essential for intelligent behavior² by altering goal priorities and generating interrupts (e.g., References 3–5).

This paper⁶ is intended to start a discussion about how the study of affect in biological systems might contribute to the development of autonomous computer systems. We suspect that from a functional perspective, some of the evolutionary forces that presumably led to the emergence of affect in animals are likely to be relevant to the design of artificial systems. However, we view this paper as only setting the stage for further research, realizing full well that it raises many more questions than it answers.

A model of affect and cognition: Three levels of behavior

In this section we outline the essence of our three-level theory of human behavior, a work that is still in progress,⁷ after which we discuss how these ideas might be applied to the development of large computer systems or computational artifacts. The ideas

©Copyright 2003 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

we discuss are still incomplete, and their implications for the design of computer systems still quite speculative. Nonetheless, we believe that even our skeleton, incomplete as it is, provides potential lessons for the design of systems that have a variety of tasks and goals, that must run unattended and autonomously, and that need high reliability. Indeed, consideration of the design constraints on autonomous robots was one of the driving forces that led to this work.⁸⁻¹³

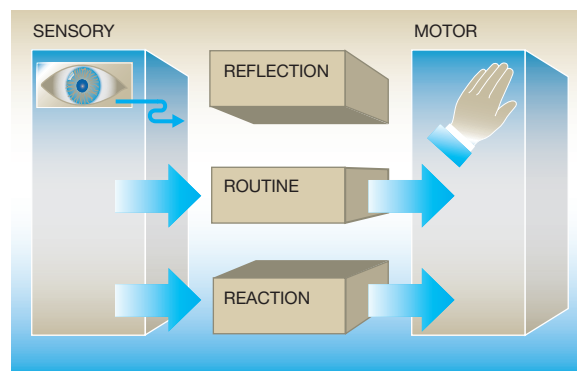
The three levels that we propose we refer to as the Reaction level, the Routine level, and the Reflection level (Figure 1). Processing at each level serves two different functions: evaluation of the world and what is happening in it—*affect*; and the interpretation of what is happening in the world—*cognition*. Higher levels involve greater depth of processing and concomitant slower processing. As shown in Figure 1, cognitive and affective information flows from level to level. Control information, in the form of activation or inhibition, flows downward.

The lowest level: Reaction. The Reaction level consists of the lowest-level processes. In animals, these processes are genetically determined and innate. No learning occurs. The Reaction level comprises immediate responses to state information coming from the sensory systems. Its function is rapid reaction to the current state.

The Reaction level monitors the current state of both the organism and the environment through fast, hard-wired detectors that require a minimum of processing. When it detects problematic or dangerous situations, it interrupts ongoing higher-level processing (if there is any), it heightens arousal, and it initiates an immediate response, or response preparation, along with a concomitant diversion of resources.

The output from the Reaction level is a set of fast and relatively simple interrupts, affective signals, and motor actions. Because of the rapid and relatively simple processing, the Reaction level cannot determine causes or do much more than respond in a simple pattern-directed manner. This level is the earliest of evolutionary processes, and in simple animals it is the only processing that occurs. In higher animals and humans, interrupts from the Reaction level trigger higher levels of processing (at the Routine and Reflection levels) in order to determine the cause and select an appropriate response. Responses at the Reaction level can be potentiated or inhibited

Figure 1 The three-level model



ited by inputs from these higher levels, and they can habituate, reducing sensitivity to expected signals.

The mid-level: Routine. In humans, the Routine level is the level of skilled and well-learned, largely “routinized” behaviors. This level is the home of most motor skills, including language generation. The Routine level is quite complex, involving considerable processing to select and guide behavior. It must have access to both working and more permanent memory, as well as evaluative and planning mechanisms. Inputs to the Routine level come from the sensory systems, the Reaction level below, and the Reflection level above in the form of control signals (inhibition and activation). The Routine level can both inhibit and activate Reaction level responses and can pass affective information up to the Reflection level when confronted with discrepancies from norms or routine expectations.

The Routine level performs assessment, resulting in values on three dimensions, which are referred to in the scientific literature on affect and emotion as positive affect, negative affect, and (energetic) arousal.¹⁴ Many emotion researchers now agree that positive and negative affect are essentially independent dimensions¹⁵ as when the motivation of a person on a diet to devour a delicious-looking cookie (a source of positive affect) coexists with the motivation to avoid the same, fattening, cookie (a source of negative affect).

As alluded to above, a key feature of the Routine level is that of default expectations. When these expectations are not met, the system can make adjustments and learn. We return to this point later in our

discussion of possible applications. But note the power of expectations in signaling potential difficulties. In humans, these expectations trigger affective processes that play an important role at the higher level of processing.

The highest level: Reflection. Reflection is a meta-process in which the mind deliberates about itself. That is, it performs operations upon its own internal representations of its experiences, of its physical embodiment (what Damasio¹ calls the “body image”), its current behavior, and the current environment, along with the outputs of planning, reasoning, and problem-solving. This level has input only

**We suggest that
affect can improve
overall systems
behavior.**

from lower levels and neither receives direct sensory input nor is capable of direct control of behavior. However, interrupts from lower levels can direct and redirect Reflection-level processing.

There is some evidence that affect changes the processing mode for cognition. The mechanism is neurochemical stimulation that adjusts the weights and thresholds that govern the operating characteristics of the cognitive mechanisms, biasing them and changing the nature of the ongoing processing. These changes influence how higher-level processing takes place, the locus of attention, and the allocation of attentional resources. Thus, negative affect, especially when accompanied by high arousal, appears to lead to more focused and deep processing—depth-first processing. In the extreme case, this type of processing leads to the “tunnel vision” of stress. In contrast, positive affect appears to lead to broad, more widely spread processing—breadth-first processing. As a result, humans have enhanced creativity when in a pleasurable state.^{16,17} Both changes are, on average, evolutionarily adaptive (one being consistent with increased vigilance, the other with increased curiosity), even if at times they are counter-productive.

Note that we propose that Reflection has only indirect control (mediated through inhibition and activation) over behavior emanating from the Routine

level. The mechanisms of this control have been explored more fully by Norman and Shallice.¹⁸

Implications for machine design

Our artificial systems today have something akin to the three different levels of Reaction, Routine (action), and Reflection, but they do not distinguish between affect (evaluation) and cognition (understanding). In this section we discuss how a model of affect and cognition along the lines of the one we have proposed might apply to machines. Specifically, we suggest that affect can improve overall systems behavior, particularly in complex or difficult environments.

The Reaction level in machines. Reaction is the home of built-in sensors, usually with prewired or preprogrammed, fixed responses. This level is necessary for safety and other critical considerations for which a rapid response is essential. The Reaction level is essential to machine operation, and indeed, is already pretty well recognized and implemented. It is common for computer systems to monitor power and temperature, hardware functioning, and checksums. In robots and other mobile systems, Reaction-level devices include contact sensors and cliff detectors that prevent the devices from hitting other objects or falling down stairs.

In animals, when dangerous conditions are noticed, not only are higher levels of processing notified, but ongoing behavior is often altered. These alterations are generally very simply implemented, and the conditions for their elicitation are easily recognized. Machines can profit even from this elementary level of adaptation to important changes in their operating environments and, as indicated above, some do.

The Routine level in machines. The Routine level is the locus of routine computational work and so involves considerable computation and reference to prior events (memory). This activity differs markedly from analyses at the Reaction level. Thus, the detection of commonplace viruses and intruders requires analysis at the Routine level. (As viruses and intruders become increasingly sophisticated, it is more likely that their detection and the corresponding remedial actions will have to be initiated at the Reflection level.)

A key feature of humans and animals is the ability to respond to deviations from norms. Consider the value for computers were they to have some mechanism for recognizing such deviations. Suppose that

as programs traversed checkpoints, they were able to detect deviations from reasonable resource and time demands and that the detection of such a deviation would trigger an alarm. In this way, excessive time (or failure) to reach a checkpoint or the use of excessive resources would trigger a search for causes and possible termination of the program. Similarly, too fast an execution or too little use of resources would signal deviant operations. We believe that capabilities of this kind would greatly enhance the reliability and dependability of our computational artifacts. These capabilities are likely to be particularly important for autonomous robots.

The Reflection level in machines. The Reflection level is the level at which the system continually monitors its own operations.¹⁹ This is both the highest level of analysis and the weakest in today's systems. Perhaps the most prevalent use of reflection is in systems that monitor such system behavior as load balance and thrashing. Reflection could lead to restructuring queues, priorities, or resource allocation. Similarly, detection of errant programs usually requires analyses at the level of Reflection. Once again, however, the automatic generation of cautionary behavior or even termination or avoidance of critical jobs does not seem to be common. Autonomous systems must have the flexibility to stop programs that could potentially lead to harm, that use excessive resources, or that appear to be in a deadlock.

Example: redundant array of independent disks (RAID). Although RAID architectures are designed to offer robust, fast access to data stored in disk arrays, along with high reliability, data are still lost. Quite often loss results from the attempt to service a disk failure.²⁰ In theory, a disk failure should do no harm, since RAID arrays are designed to handle this contingency: the failed drive is pulled out and a good one put in. But occasionally the operator swaps out the wrong one, causing a second failure, and so data are lost.

There are a couple of approaches available to reduce data loss. One would be to make the RAID safe, even with two failures (e.g., RAID-6). A second would be to design the interface better to minimize such errors. This approach is clearly better: the value of efficient human-computer interaction is well-known, albeit too-seldom practiced. But the first approach comes at a price, namely, increased cost and loss of efficiency. Here is where the affective system would be useful.

Suppose that the loss of a disk drive is detected at the Reaction level and used to trigger an alert: in essence, the system would become "anxious." Yes, the human operator would be summoned, but here

**Lack of warning
is a common
problem in automated
systems.**

the Routine level would kick in, retrieving past instances where service by the human operator had led to increased problems: this would serve to increase the anxiety level. The result of this increased anxiety would lead to an operations change—to a more conservative approach implemented by a change in policies. Because the margin of safety has been lowered, the system could institute more frequent checkpoint saves, perhaps to a remote location (after all, the RAID is no longer fully trustworthy), and perhaps the system could run a parallel shadow operation or postpone critical jobs. An alternative operation would be to restructure the RAID on the fly to make it tolerate further disk failure without damage, even at the cost of decreasing its capacity or slowing its operation.

In other words, why should computer systems not be able to behave like humans who have become anxious? They would be cautious even while attempting to remove the cause. With humans, behavior becomes more focused; they tend to engage in in-depth problem-solving first until the cause and an appropriate response are determined. Whatever the response for machine systems, some change in normal behavior is required.

Lack of warning is actually a common problem in automated systems.²¹ The systems are well-designed to function even in the case of component failure, but they seldom report these failures to higher-level systems or change their behavior. As a result, the human operator, or higher-level monitors of the system, may be unaware that any problems have occurred even though error tolerance is now much reduced. Occasionally, further failures carry the system over the threshold of recoverability, often leaving the human operator to cope with the resulting unexpected emergency.

If systems followed the human model of affect, all failures would be reported, and just as in a person, a rising level of anxiety would trigger a change in focus and behavior at higher levels, preparing for eventual disaster and thereby minimizing its impact or possibly avoiding it altogether.

Why use affect? Why not just program the system to safeguard itself against problems? For any specific problem that might arise, once that problem is known and understood, the most effective solution will always be to write an appropriate algorithm to deal with it. So why are we proposing the introduction of a new system, that of affect? Why not simply analyze each potential failure and deal with it efficiently?

Normally, when thinking about computer systems design, we think in terms of what in artificial intelligence are referred to as *strong methods*, that is, methods that exploit specific domain knowledge and structure. In other words, we think in terms of specific algorithms that solve specific problems by incorporating substantial knowledge about the problem into the algorithm. By contrast, *weak methods* and *heuristics* do not incorporate domain knowledge because they are designed to be much more general. The result is that they are generally much slower, much less efficient, and often are not guaranteed to succeed. Weak methods trade efficiency for generality. Thus, for example, hill climbing is a weak method that has great generality, but is often inefficient and can become trapped by local maxima.

Strong methods are always preferable when the situations are known and understood and the environment predictable and relatively limited in scope. But when these conditions do not hold, weak methods are preferable. Affect is a computationally weak method. Its power lies in its capacity to help deal with unexpected problems, so that it complements strong, algorithmic methods by adding robustness in unanticipated situations. The real world is characterized by uncertainty and variability. For these cases, biology uses weak methods—methods that are general and applicable to a wide variety of situations. As machines become more autonomous and more exposed to uncertainty, affect will become an increasingly appropriate solution for them as well.

Biology, of course, is not without its strong methods. Even humans with their big brains have retained numerous wired-in, efficient responses to particular situations. Reflexes and tropisms respond rapidly to

particular stimulus conditions such as lack of support, unbalance, bitter taste, the smell of putrefaction, and hot or sharp surfaces. These responses are rapid, pattern-driven solutions to specific classes of events. But biology also uses more complex, slower, reflective problem-solving and planning to deal with novel situations. Thus, biological systems make rapid responses to situations that require them, and slow, considered responses when circumstances demand them and time permits.

Implications

An affective computer would be able to sense the state of its own operations and that of its environment. It would be able to compare its behavior with its expectations, and it would be able to reflect upon its own operations. It would have knowledge about its own trustworthiness and about that of the other systems with which it interacts, and it would be able to modulate its overall behavior toward better performance by sensing things that are not now taken into account, acting cautiously where appropriate and aggressively where possible. It would automatically reconfigure itself to take account of increased risk and would continuously be aware of the state of its own health, at least from an infrastructure and computational point of view.

We propose that by continually sensing its own state and that of its environment, the system would essentially be controlling its level of satisfaction or anxiety. When components needed service, the level of anxiety would rise, for the need for service means that error tolerances are lowered and the very act of service can cause errors. Just as human operators know not to do system maintenance or a software upgrade during or just before some critical job needs to be performed, so computer systems themselves should have the same sense of anxiety.

Imagine a grid computer, assembling a number of machines prior to doing a computation. Suppose that each machine were queried about its state of readiness, in essence asking “How are you feeling?” The range of possible responses given below is instructive:

“I had a disk failure in my RAID, so if this is an important calculation, you had better not count on me.”

“I am feeling a bit anxious because I have had some errors, so I will be slowed by the need to do con-

tinual checks.” (This response shows how a machine might provide a graded level of service.)

“I am feeling anxious because of recent virus or hacker attacks.”

Animals have developed sophisticated mechanisms for surviving in an unpredictable, dynamic world, coupling the appraisals and evaluations of affect to methods for modulating the overall system. The result is increased robustness and error tolerance. Designers of computer systems might profit from their example.

Acknowledgments

We thank Bill Revelle, Ian Horswill, and Tony Tang of Northwestern University for their contributions to the general theory and to the particular recommendations made here.

Cited references and notes

1. A. R. Damasio, *Descartes' Error: Emotion, Reason, and the Human Brain*, G. P. Putnam, New York (1994).
2. M. Minsky, *The Emotion Machine*, Pantheon, New York, forthcoming.
3. N. Frijda and J. Swagerman, “Can Computers Feel? Theory and Design of an Emotional System,” *Cognition & Emotion* **1**, No. 3, 235–257 (1987).
4. H. A. Simon, “Motivational and Emotional Controls of Cognition,” *Psychological Review* **74**, 29–39 (1967).
5. A. Sloman and M. Croucher, “Why Robots Will Have Emotions,” *Proceedings of the Seventh International Conference on Artificial Intelligence* (1981).
6. Originally presented at the IBM Autonomic Computing Summit at the Thomas J. Watson Research Center, May 14–15, 2002.
7. D. A. Norman, A. Ortony, and W. Revelle, “Effective Functioning: A Three Level Model of Affect, Behavior, and Cognition,” in *Who Needs Emotions? The Brain Meets the Machine*, J. M. Fellous and M. A. Arbib, Editors, to be published.
8. There is beginning to emerge a substantial body of literature^{9–12} on “affective computing,” systems designed to recognize or simulate human affect. Much of this work, however, does not deal with the design of machine architectures. Our own views have been particularly influenced by work that does, particularly that of Aaron Sloman.^{11,13} In common with Sloman, we propose that human information processing operates at three levels. Our three levels, the Reaction, the Routine, and the Reflection levels are related to, but somewhat different from those of Sloman (see Reference 13). In particular, whereas his reactive level is essentially the same as our Reaction level, his “deliberative reasoning” is related to but different from our “Routine” level, and his “meta-management” level is similarly related to but somewhat different from our Reflection level. Other differences are not relevant to this discussion.
9. C. Breazeal, *Designing Sociable Robots*, MIT Press, Cambridge, MA (2002).
10. R. W. Picard, *Affective Computing*, MIT Press, Cambridge, MA (1997).
11. A. Sloman and B. Logan, “Evolvable Architectures for Human-Like Minds,” *Affective Minds*, G. Hatano, N. Okada, and H. Tanabe, Editors, Elsevier, Amsterdam (2000), pp. 169–181.
12. R. Trappl, P. Petta, and S. Payr, *Emotions in Humans and Artifacts*, MIT Press, Cambridge, MA (2003).
13. A. Sloman, “How Many Separately Evolved Emotional Beasts Live within Us?,” *Emotions in Humans and Artifacts*, R. Trappl, P. Petta, and S. Payr, Editors, MIT Press, Cambridge, MA (2003).
14. R. E. Thayer, *The Biopsychology of Mood and Arousal*, Oxford University Press, New York (1991).
15. J. T. Cacioppo and W. L. Gardner, “Emotion,” *Annual Review of Psychology* **50**, 191–214 (1999).
16. F. G. Ashby, A. M. Isen, and A. U. Turken, “A Neuropsychological Theory of Positive Affect and Its Influence on Cognition,” *Psychological Review* **106**, No. 3, 529–550 (1999).
17. A. M. Isen, “Positive Affect and Decision Making,” *Handbook of Emotions*, M. Lewis and J. M. Haviland, Editors, Guilford, New York (1993), pp. 261–277.
18. D. A. Norman and T. Shallice, “Attention to Action: Willed and Automatic Control of Behavior,” *Consciousness and Self Regulation: Advances in Research*, Vol. IV, R. J. Davidson, G. E. Schwartz, and D. Shapiro, Editors, Plenum Press, New York (1986).
19. Our use of “reflection” is related to the sense intended in computational reflection, whether in programming languages or operating systems. Both uses emphasize the capability of a system to examine its own operations, but the details and goals differ.
20. A. Brown and D. A. Patterson, “To Err Is Human,” *Proceedings of the First Workshop on Evaluating and Architecting System Dependability (EASY '01)*, Göteborg, Sweden (July 2001).
21. D. A. Norman, “The ‘Problem’ of Automation: Inappropriate Feedback and Interaction, Not ‘Over-Automation,’” *Human Factors in Hazardous Situations*, D. E. Broadbent, A. Baddeley, and J. T. Reason, Editors, Oxford University Press, Oxford (1990), pp. 585–593.

Accepted for publication September 26, 2002.

Donald A. Norman Department of Computer Science, Northwestern University, 1890 Maple Avenue, Evanston, Illinois 60201 (electronic mail: norman@northwestern.edu). Dr. Norman is Professor of Computer Science at Northwestern University and cofounder of the Nielsen Norman group where he serves on the advisory boards of numerous firms. He is also Professor Emeritus of Cognitive Science and Psychology at the University of California, San Diego. Dr. Norman is the author of numerous books, including *Design of Everyday Things* and *The Invisible Computer*. He has served as Vice President of Advanced Technology at Apple Computer, Inc. He has B.S. and M.S. degrees in electrical engineering and a Ph.D. degree in experimental psychology.

Andrew Ortony Department of Psychology and the School of Education and Social Policy, Northwestern University, 2120 Campus Drive, Evanston, Illinois 60201 (electronic mail: ortony@northwestern.edu). Prof. Ortony is a professor of psychology and education, and codirector of Northwestern University's Center for the Study of Cognition, Emotion, and Emotional Disorders. His primary research interests concern the relation between emotion, cognition, behavior, and personality, and the implications

of these for artificial intelligence (AI) and for interface design. His coauthored book *The Cognitive Structure of Emotions* is a widely used resource in AI applications involving emotions. Prof. Ortony is a Fellow of the American Psychological Association, the American Psychological Society, and a member of the American Association for Artificial Intelligence, the Cognitive Science Society, and the International Society for Research on Emotion.

Daniel M. Russell *IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, California 95120 (electronic mail: daniel2@us.ibm.com).* Dr. Russell is the senior manager of the User Sciences and Experience Research (USER) lab at the Almaden Research Center. The main interests of the laboratory are in the areas of designing the complete user experience of computation, especially in the domains of highly sensed or attentive environments, formalizing the characteristics of human behaviors for input mechanisms, and creating new ways of placing computation into the work space. Prior to coming to IBM, Dr. Russell worked at Xerox PARC and in the Advanced Technology Group at Apple Computer, Inc. He founded and managed the User Experience Research (UER) groups at both companies. He received his B.S. in computer science from the University of California, Irvine, in 1977 and his M.S. and Ph.D. degrees from the University of Rochester in 1979 and 1984, respectively.